Quantifying the reliability of fault classifiers

Olga Fink, Enrico Zio, Ulrich Weidmann

▶ **To cite this version:**

Olga Fink, Enrico Zio, Ulrich Weidmann. Quantifying the reliability of fault classifiers. Information Sciences, 2014, 266, pp.65-74. 10.1016/j.ins.2013.12.008 . hal-00930985

HAL Id: hal-00930985

https://hal-centralesupelec.archives-ouvertes.fr/hal-00930985

Submitted on 14 Jan 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Quantifying the reliability of fault classifiers

Olga Fink[a,*], Enrico Zio[b,c], Ulrich Weidmann[a]

[a] *Institute for Transport Planning and Systems, ETH Zurich, Zurich, Switzerland*
[b] *Chair on Systems Science and the Energetic Challenge, European Foundation for New Energy-Electricité de France (EDF) at École Centrale Paris and SUPELEC, France*
[c] *Department of Energy, Politecnico di Milano, Italy*

Abstract

Fault diagnostics problems can be formulated as classification tasks. Due to limited data and to uncertainty, classification algorithms are not perfectly accurate in practical applications. Maintenance decisions based on erroneous fault classifications result in inefficient resource allocations and/or operational disturbances. Thus, knowing the accuracy of classifiers is important to give confidence in the maintenance decisions. The average accuracy of a classifier on a test set of data patterns is often used as a measure of confidence in the performance of a specific classifier. However, the performance of a classifier can vary in different regions of the input data space. Several techniques have been proposed to quantify the reliability of a classifier at the level of individual classifications. Many of the proposed techniques are only applicable to specific classifiers, such as ensemble techniques and support vector machines. In this paper, we propose a meta approach based on the typicalness framework (Kolmogorov's concept of randomness), which is independent of the applied classifier. We apply the approach to a case of fault diagnosis in railway turnout systems and compare the results obtained with both extreme learning machines and echo state networks.

Keywords: Confidence estimation, Reliability of classifiers, Typicalness framework, Railway turnout system, Extreme learning machines, Echo state networks

1. Introduction

The potential impact of failures and malfunctions of components and systems of hazardous plants and critical infrastructures has motivated an increased use of monitoring devices for operation, control and condition-based maintenance. Monitoring data provide information on the state of components and systems, which can be used for fault detection and diagnostics [7]. If an impending or an incipient failure condition can be detected, isolated and identified [30], [23], the operator can proactively intervene to prevent an interruption of operation.

We consider fault diagnostics, for which there are different approaches. Data-based approaches in particular have been emerging as an effective solution in practical applications [18], [32], [28]. The task of fault diagnostics can be formulated as a classification problem: separating the data patterns in the input space into distinct classes [1]. To solve this task corresponds to finding a separating hyperplane, respectively a decision boundary that separates the data into different classes as accurately as possible, i.e. with minimum classification error [9].

Classification tasks can be learned either in a supervised or an unsupervised way. In this paper, we focus on supervised learning, for which the mapping between input patterns and target labels is known. The task of the classification algorithm in supervised learning is to deduce the implicit relationship between the patterns in the input data and the target labels [1]. The separating hyperplane is found through a learning process driven by minimization of a defined loss function, e.g. the average number of misclassified patterns. The trained classifier is, then, used to predict the labels of unknown patterns different from those of training.

For effective maintenance decision-making, classifications need to be provided with a measure of their accuracy. In supervised learning, the accuracy of a classifier has been measured by means of the confusion

*Corresponding author; Email: ofink@ethz.ch; Tel: +41 44 633 27 28; Postal Address: ETH Zurich, Wolfgang-Pauli-Str. 15, 8093 Zurich, Switzerland

matrix or graphical methods, such as Receiver Operating Characteristic (ROC) curves [17]. Additionally, information theoretic measures have been applied [19].

Most of the measures of learning algorithms quantify performance on average over all the patterns in a given testing dataset or in different folds by cross-validation [17]. The average performance is a good indicator for the selection of a classification algorithm and for setting its parameters optimally. However, when using the classifier for fault diagnostics based on a specific pattern of monitored data, the pattern could lie in a region in the input space which has not been sufficiently densely covered in the training dataset, or it could even be an outlier in the input space. In these cases, even though the average performance of the algorithm may have been validated as high, the classification algorithm may still perform poorly. In practice, then, the measure on the average performance of a fault classifier may not be sufficiently informative for the maintenance decision maker. Therefore, reliability measures for individual classifications are required to support the decision maker and prevent that decisions are based on unreliable classifications.

There can be several reasons for poor classification results. One reason can be that the incoming pattern is drawn either from a region in the input space which has not been sufficiently covered by patterns during training of the classification algorithm or from a region which is close to the separating hyperplane of the discriminant function built by the algorithm, where the patterns of different classes are very similar and cannot be easily distinguished by the algorithm. Another reason of poor classification performance can be due to the characteristics and structure of the algorithm itself, that may not be capable of learning the patterns of the training dataset adequately. Yet another reason of poor performance could be that the incoming pattern is an outlier, dissimilar to all the patterns of the training dataset.

While for regression tasks, confidence intervals can be assigned to each of the predicted values, the confidence of a classifier is more difficult to assign because of the discrete character of the classification task, whereby the output of the classification algorithm can be either correct or not. Various approaches have been proposed in the literature to quantify the confidence in individual pattern classifications. Most have been designed for a specific classifier. For example, the degree of agreement, respectively the disagreement, of individual neural networks within an ensemble classifier can be used as indicator of the confidence in the individual classifications [11].

The concept of typicalness of patterns has been proposed by Vovk et al. [31] to assign a confidence and a credibility measure to an individual classification. In addition to the specific approach of support vector machines, the design of a nearest-neighbor classifier based on the typicalness of the specific pattern has been proposed by Gammerman and Vovk [8].

Smirnov et al. [29] extends the typicalness approach so that it can be applied to assess the performance of an arbitrary classifier. The proposed meta approach comprises an arbitrary classifier that performs the main classification task and a meta classifier that predicts the reliability of each single classification. The output of the first classifier is used to train the meta classifier.

Even though Barbara et al. [3] do not use the typicalness approach to assign confidence, respectively reliability values to patterns, they use the approach to construct an outlier detector by unsupervised learning.

In this paper, an approach is proposed that extends and combines the previously proposed approaches based on typicalness. A two-step meta classifier is proposed. The classifier based on the typicalness of the neighboring patterns, proposed by Gammerman and Vovk [8], is used solely to quantify the reliability of the primary classifier, which is applied for the actual classification task. The typicalness values based on the distance to the nearest-neighbors are thereby not used as the sole input to the classifier but are complemented by the typicalness value defining the outlier character of the pattern, similar to the one proposed by Barbara et al. [3]. Additionally, also the "confidence" of the primary classifier in the performed prediction is used as input to the meta classifier. The meta approach enables the usage of the best performing classifier for the main classification task and the quantification of the reliability of that classifier based on aggregated information of the input space and on the performance of the main classifier conditional on the presented input patterns. Contrary to this approach, the meta classifier approach proposed by Smirnov et al. [29] uses the original input as input to the meta classifier. The approach proposed in this paper improves the generalization ability and the accuracy by integrating several indicators of typicalness. The proposed approach is applied to a degradation classification problem from a railway turnout system. The diagnostic classification task is performed by two different algorithms: extreme learning machines and echo state networks. Subsequently, the results of quantifying the reliability of the two classifiers are compared.

The remainder of the paper is organized as follows. The next Section of this paper first defines the terms applied within the proposed framework and differentiates them from those applied in some other studies.

Subsequently, the background of the typicalness framework is presented and the proposed framework, the applied typicalness values and the proposed procedures are introduced. In Section 3, the proposed approach is applied to the case study of a railway turnout system. In Section 4, the obtained results are discussed. Finally, Section 5 presents the conclusions of this research.

2. Meta approach to quantify the reliability of classification

2.1. Terminology

Before the proposed approach and the underlying theory are presented, some key terms used in this paper are introduced and differentiated from those applied in some other studies.

The term *reliability of a classifier* has been used in several studies [29], [5]. Partly, the term has been used as a superordinate qualitative performance property, comprising different aspects of performance assessment indicators, such as accuracy, confidence or credibility [5]. The term *reliability estimate* was used to quantify different aspects of reliability. In other studies, the term *reliability* has been used quantitatively to express the probability that the classifier performs the prediction correctly [29]. In this study, the term *reliability* is used according to this latter interpretation.

There are several other performance indicators. Gammerman and Vovk [8] introduced the indicators of confidence and credibility, related to the typicalness approach. The confidence in this context is defined as one minus the second largest randomness level. This describes the probability that the pattern has not been classified as the second typical class but as the most typical class. The higher the value of confidence, the higher the untypicalness of the specific pattern with respect to other patterns in the second typical class and the more certain therefore the mapping to the most typical class.

The credibility is defined as the randomness level of the output prediction. Therefore, the accuracy of a prediction is determined by the combination of the two indicators: the higher the credibility and the confidence, the higher the accuracy of a classification. One of the indicators is in this case not sufficient to represent the accuracy.

The term confidence has been, partly, used differently, in other studies. Baraldi et al. [2] have defined the level of confidence as the probability that the assigned class is correct, given the specific pattern.

In our work based on the typicalness approach, we adopt the definitions *of confidence and credibility* introduced by Gammerman and Vovk [8]. The term *reliability* is used to quantify the probability that the performed classification is correct.

2.2. Basics of the typicalness approach

The proposed meta approach comprises two classifiers: a main (primary) classifier and a meta classifier. While the main classifier is trained to generalize the patterns of the original input space, the meta classifier is trained to recognize the levels of reliability of the main classifier based on three different typicalness values.

The idea of using the typicalness to quantify the classifier reliability is based on the consideration that if the considered pattern is typical compared to the patterns that have already been classified correctly by the algorithm during the training process, it is expected that the performed prediction would be reliable. However, if the pattern is very untypical then the performed prediction is expected not to be reliable.

The typicalness of a pattern can be defined with respect to several aspects. The definition at the basis of our approach uses Kolmogorov's concept of randomness [21]. A sequence is considered random if it cannot be compressed (if there are no repeating patterns). Kolmogorov complexity provides a universal randomness test but it is not computable [21]. It has been extended by Martin-Löf [25] for practical applications. Generally, finite sequences are not solely random or non-random but have different degrees of randomness. To measure this, a concept of randomness deficiency is introduced based on the *iid* assumption (identically and independently distributed) [21].

Given Z, the set of all possible labelled examples and Z^*, the set of all finite sequences of labelled examples, a function $f : Z^* \to [0, \infty)$ is a *randomness test* if:

(1) for all $r \geq 0$, all $n \in \{1, 2, \ldots\}$ and all probability distributions P in Z, $P^n\{z \in Z^n : f(z) \leq r\} \leq r$;
(2) f is upper semicomputable.

The first condition defines that the randomness test is required to be valid while the second condition defines that the test should be computable, in some weak sense.

Even though the randomness level is not computable, it can be approximated by various approaches. For example the Lagrange multipliers of the optimization problem in support vector machines can be used to express the "strangeness" of a solution [8]. Generally, the level of randomness can take values between 0 and 1, with a 0-level of randomness being absolutely untypical and a level of 1 being very typical.

In addition to the use of Lagrange multipliers, the approximation of the strangeness values α_i can also be obtained by a nearest-neighbor algorithm [8]:

$$\alpha_i^{nn} := \frac{\sum_{j=1}^{k} d_{ij}^{+}}{\sum_{j=1}^{k} d_{ij}^{-}}, \tag{1}$$

with $\alpha_i^{nn} \in \mathbb{R}$, where d_{ij}^{+} is the shortest distance from x_i to the jth nearest neighbor classified in the same class as x_i, d_{ij}^{-} is the shortest distance from x_i to the jth nearest neighbor classified differently from x_i, and k is the number of nearest neighbors.

The rationale behind definition (1) is that the shorter the distance to the patterns from the same class and the larger the distance from the patterns classified differently, the smaller is the value of α_i^{nn} and consequently, the less strange is the pattern with respect to its neighbors. However, if the pattern is classified differently than its neighbors, then, its distance to the patterns of the same class is larger than the distance from the patterns of other classes and α_i^{nn} will be larger than 1.

With the computed strangeness values, the typicalness value, p, can be computed:

$$p(y_{n+1}) = \frac{\#\{i : \alpha_i \geq \alpha_{n+1}\}}{n+1}, \tag{2}$$

with $p(y_{n+1}) \in \mathbb{Q}$.

Equation 2 expresses the proportion of α_i which are at least as large as the last α_{n+1}, with n being the number of training patterns. Therefore, always the strangeness value of one pattern at a time is compared to the strangeness values of the patterns used to train the classification algorithm. The larger the strangeness value of the considered pattern y_{n+1} is, the fewer strangeness values from the training dataset will be larger than that of the considered pattern and consequently, the less typical the pattern will be with respect to those contained in the training dataset.

Because the typicalness approach is solely based on the *iid* assumption it is flexibly applicable to arbitrary input distributions.

2.3. Definition of typicalness values

There are different parameters with respect to which the typicalness of a pattern can be described. For the proposed meta approach, three different typicalness measures are taken as inputs, each of them based on a source of potential misclassifications. For each of these potential sources a strangeness measure α_i and a pertinent typicalness value p_i are assigned. The different typicalness inputs are required to enable the algorithm to integrate different typicalness aspects of one pattern and to classify it accordingly. Furthermore, if considered separately, single typicalness values would result in many false positives, respectively false negatives. The combination of several typicalness characteristics enables the meta classifier to discriminate the poorly classified patterns based on several aspects.

For the typicalness of the input patterns, the approach based on nearest-neighbors is selected. First, the strangeness, α_i^{nn}, is computed based on Equation 1. Subsequently, the typicalness of the patterns is computed based on Equation 2. The strangeness expresses the ratio of the distances of the specific pattern to the nearest neighbors from the class to which it has been classified, to the distances to the nearest neighbors of all other classes. The strangeness is the smaller the nearer the pattern is to the patterns from the same class and the more distant it is to the patterns of other classes. If the pattern is in the region of the separating hyperplane between several classes, the strangeness value approaches 1 as the pattern is similarly distant to the same class patterns as to the patterns from other classes.

To characterize the outlier property of a pattern, a further strangeness approach is introduced, similar to the one introduced by Barbara et al. [3]. At first sight, the strangeness based on the nearest-neighbors approach appears to cover also the outlier character of the pattern as the distance to the nearest neighbors is measured and the distance increases with the outlier character of the pattern. However, Equation 1

represents the ratio of the distance to the nearest neighbors within the same class, to the nearest neighbors of all the other classes. If a pattern is an outlier, its distance both to the class to which it has been classified and to all other classes is large. The ratio between these two numbers can be one, and in this case the patterns are similarly distant from all the classes; or it can be smaller than one, in which case, even though the distances are large, the ratio between them becomes a number that is comparable to other levels of randomness; or it can be bigger than one, in which case the distances between the pattern class and other classes are not similar. Therefore, the strangeness indicator α_i^{nn} is not always able to reflect the outlier character of a pattern and can have a similar degree of randomness as patterns that do not have an outlier character. Consequently, an additional indicator, α_i^{out}, is introduced to express the randomness of the patterns with respect to the outlier character, (Equation 3). In this case, only the distance to nearest neighbors of the same class is considered.

$$\alpha_i^{out} := \sum_{j=1}^{k} d_{ij}^{+}, \tag{3}$$

with $\alpha_i^{out} \in \mathbb{R}$.

α_i^{out} complements the randomness indicator α_i^{nn}, which is based on the ratio between the distances to the same class and the distances to all other classes (Equation 1).

For this strangeness indicator, the rationale is that the larger the distance is from the considered pattern to those from the same class, the larger the strangeness is of the considered pattern. Consequently, the more similar the pattern is to those from the same class, the smaller the distance is to these patterns and the less strange the pattern is.

If, for example, α_i^{out} is large and α_i^{nn} is approximately one, it can be concluded that the pattern has an outlier character and is equally distant to the patterns in all the classes.

To characterize the typicalness of the output, a different indicator is introduced. There are, generally, several approaches to design classifiers. One approach is to make the output one-dimensional so that the output can be either binary for a two-class classifier or discrete for multi-class classifiers. For scoring classifiers it is also possible to define a cutoff value and to define the classification decision rules in this way. The cutoff values can be varied based on the performance of the classifier on the critical classes. Another approach is to set the output dimension to the number of possible classes and to represent the membership of the class by a binary number. However, if the output variable is not binary but continuous, then for a selected pattern the membership to a class is defined by the maximum score of all possible classes. By this approach, a continuous variable is converted to a binary variable. Thereby, the information is lost about how "certain" the classification algorithm was of the specific classification. Therefore, even though the assigned class might have had the maximum value out of the outputs of all the classes, the difference to the output values of other classes, nevertheless, might have been small. Consequently the output would be less certain in comparison to the case in which the output of the assigned class is equal to one and that of all the other classes equal to zero, respectively -1.

Based on these considerations, the output can be directly used to define the strangeness of the algorithm performance within the typicalness framework. Consequently, one minus the maximum output of the algorithm for each pattern is directly defined as the strangeness value, α_i^{alg}:

$$\alpha_i^{alg} := 1 - max(\widehat{y}_{i1}, \ldots, \widehat{y}_{ij}), \tag{4}$$

where $\widehat{y}_{ij}$ is the output of the algorithm for the class j for the ith pattern. In case of a binary classifier, the maximum value of only two outputs $\widehat{y}_{i1}$ and $\widehat{y}_{i2}$ is taken. The smaller the maximum output value $\widehat{y}_i$, the more "uncertain" the classifier is about the computed output and more untypical is the value. Considering a binary classifier, in the ideal case the maximum value $\widehat{y}_i$ will be equal to 1 and the resulting strangeness value will be 0. In the worst case, the algorithm will not be able to distinctly distinguish between the two classes and the maximum value of $\widehat{y}_i$ will approach 0.5 and the strangeness value, consequently, will also approach 0.5.

To define the typicalness values p_i^{alg}, again, Equation 2 is applied.

2.4. General meta approach

The proposed meta approach is composed of two classifiers: a main classifier and a meta classifier. The first classifier performs the main classification task. The second classifier predicts the level of reliability

of the classification performed by the main classifier. The main classifier is based on the original input space whereas the meta classifier uses typicalness values of the original input and the output of the main classifier to learn to predict the misclassified patterns without having the information on the target label of the specific pattern. The specific algorithms used within the single classifiers can be selected according to the classification task and the specific requirements.

In the first step, an arbitrary classifier is applied to perform the general classification task. Holdout technique with a training and a testing datasets is applied to validate the performance of the classifier. Subsequently, randomness values, α_i^{nn}, α_i^{out}, α_i^{alg}, and the pertinent typicalness values, p_i^{nn}, p_i^{out}, p_i^{alg}, are computed. Additionally, target output, y^{meta} for the binary meta classifier is computed based on the accuracy of the main classifier on the training dataset: if the pattern has been correctly classified, *class 1* is assigned, otherwise *class 2* is assigned. Next, the meta classifier is trained to distinguish between the patterns classified correctly and incorrectly by the main classifier. Subsequently, the strangeness values α_i^{meta} and the typicalness values p_i^{meta} are computed similarly to α_i^{alg} and p_i^{alg} based on the typicalness values and the target output. The typicalness of the output of the meta classifier can be defined as the reliability of the classification and defines the probability that the predicted class is correct. The threshold for unreliable classifications is defined as the maximum p_i^{meta} value. With this threshold, the patterns are determined for which the classification is expected to be unreliable. The different steps of the proposed framework are displayed in Figure 1.

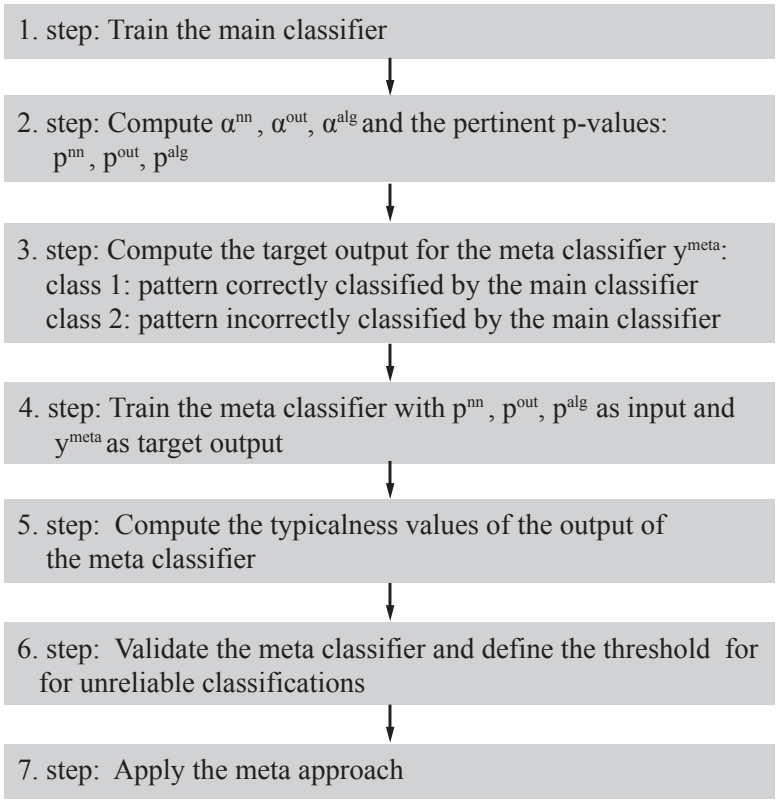

Figure 1: Framework of the meta approach

3. A case study of classification of faults of railway turnout systems

3.1. Applied procedure and algorithms

The proposed approach is applied to a case study of railway turnout systems fault diagnosis, which is defined as a classification task. All the steps of the proposed framework (Figure 1) are applied to a real dataset.

For quantifying the randomness values for which a determination of nearest neighbors is required, the balltree data structures are used to enable efficient searches in high-dimensional spaces [27]. The distances between the patterns are computed as Euclidean distances.

For the main classifier, two different classifiers were tested: extreme learning machines (ELM) and echo state networks (ESN). For the meta classifier, ELM were used.

These algorithms have been selected due to their good performance on several benchmark studies [15], [16], the simplicity of their parameter setting and their flexibility.

The ELM is a feedforward neural network with a single hidden layer and flexible processing units [14]. ELM combines the strengths of several machine learning techniques, such as support vector machines with kernels, but also feedforward neural networks with different activation functions, such as linear, sigmoidal, polynomial and radial-basis functions [13]. The learning algorithm of ELM not only combines these activation functions within the hidden processing units, but also enhances the state-of-the-art approaches by speeding up the learning process of the algorithms and by avoiding local minima, which is one of the major drawbacks of gradient-based learning algorithms [15]. Several extensions to ELM have been introduced [20], [6], [26].

The parameters of the ELM do not have to be set and tuned manually, but are either set randomly or determined within the learning procedure.

ESN are a specific type of recurrent neural networks [24]. Similar to other recurrent neural networks, ESN are able to exhibit dynamic temporal behavior and have a memory [16]. ESN are typically applied for modeling complex dynamic systems.

The main advantages of ESN are their efficient learning, the dynamics and the memory, and particularly, the flexibility with respect to the application and the possible combination with other powerful algorithms. Contrary to other neural networks in which the neurons are organized in layered structures, ESN comprise a reservoir as the main structural element, in which the neurons are randomly and sparsely connected [10]. The weights between the connected neurons within the reservoir are fixed and are not trained during the training process. Only the weights between the reservoir and the output are determined by (ridge) regression, which is computationally inexpensive compared to the backpropagation learning procedure [22].

The parameters of the ESN algorithm were set by hyperparameter optimization performed [4].

Ridge regression [12] with a regularization term of 0.01, which imposes rigidity, was applied in all the classifiers.

3.2. Analysed system and applied data

Turnouts are critical components within the railway network. They enable trains to be guided from one track to another and consist of several parts including turnout blades, stock rails, the so called “frog” and the turnout actuator which positions the moveable parts of the turnout (Figure 2).

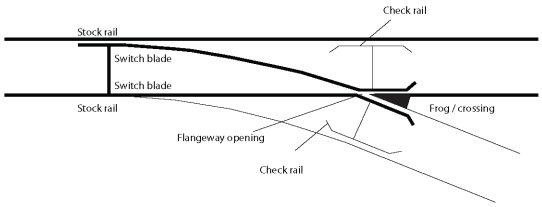

Figure 2: Parts of the turnout system

As with all physical infrastructure systems, turnout systems are subject to degradation. The turnout degradation process depends on several parameters, including axle loads, train speeds, conditions of the train wheels, and environmental conditions.

Sensors and other devices are increasingly used to monitor the condition and performance of railways infrastructure and operations. This is especially true for components like turnouts and particularly for those in critical locations, which have high capacity utilization rates. By anticipating component failures, the monitoring devices help railway operators plan and implement cost effective maintenance regimes.

The data used in this case study were collected from six force measurement devices installed along the turnout blades and the frog components of a turnout system installed in a railway tunnel portal. The force

measurement is activated when the positioning process starts and the system records the forces applied for each positioning at several locations along the turnout. The system measures the applied forces for each millisecond of the positioning process. In the post-processing of the data, the system also computes the work performed by the actuator system and stores this information for each of the measurement locations separately. The performed work corresponds to the integral of the force curve. Since the turnout is positioned in different directions, the applied forces can vary. For this case study, positioning processes for only one direction were considered. The total observation period considered in this research was about 3.5 years.

3.3. Classification

For the classification task, two levels of aggregation were considered. On the disaggregated level, distinct force-curves for one single selected movement mechanism were evaluated. To classify these distinct force-curves, the aggregated work performed by all the six monitoring points along the turnout was applied. Two classes were defined: force-curves with an overall high ("high-class") or low level of performed work ("low-class"). The classification task aimed to demonstrate that the features of the shapes of force-time curves are sufficiently distinguishable between the patterns with an overall high level and low level of performed work, without taking the absolute value range in consideration. Therefore, each force-curve was normalized in the interval $[0, 1]$ with the distinct value range of each curve.

In total there were $11,966$ patterns in the dataset, $6,159$ of which belonged to the high-class and $5,809$ to the low-class. In the first step, random re-sampling without replacement [17], with apportionments between 10% and 90% of the dataset between training and testing, was used to validate the performance of the ELM algorithm and the robustness to variations in different input datasets. The dataset was randomly partitioned into training and testing datasets, according to the defined percentage of data allocated for training. Contrary to alternative cross-validation approaches, such as leave-one-out or k-fold cross-validation, random re-sampling does not guarantee that each of the samples will be selected for the testing dataset. However, this approach is more flexible with respect to different apportionments of the training and testing data. The re-sampling was repeated 100 times for each of the defined apportionments of data allocated for training and the average classification accuracy of the algorithm on the testing datasets was computed. The results are shown in Figure 3. It is shown that going from using 90% of data for training to only 10% resulted in a performance drop of only about 0.7% in the average classification accuracy on the testing data. The classification performance also showed to be robust to variations in different input datasets.

The increasing variance of the performance on datasets with increasing apportionments to the training dataset can be explained by the decreasing sample size of the testing dataset and the increasing influence of single misclassifications on the performance of the algorithm. For example in case of the 90%-training dataset, 8.4 patterns are misclassified on average and one single additional misclassified pattern influences the performance by 0.08%.

For further considerations, the holdout technique was applied to validate the average performance of the algorithm. The training dataset contained 90% of the entire available dataset and the testing dataset the rest of it ($10,769$ patterns were used for training and $1,197$ for testing).

The ELM algorithm classified 99.70% of the training data patterns and 99.33% of the testing data patterns correctly. In the testing dataset, eight patterns in total, out of $1,197$, were misclassified. There was a very small discrepancy between the training and testing error, which is an indication of a very good generalization ability of the algorithm. A similar classification accuracy for both classes was observed, with 99.49% of the patterns from the high-class and 99.17% of the low-class being classified correctly. This corresponds to 3 patterns from the high-class and 5 patterns from the low-class being misclassified, out of 8 misclassified patterns in the testing dataset in total.

The ESN algorithm showed a slightly worse performance classifying correctly 99.43% of the training patterns; on the testing dataset, a similar performance was obtained, with only one additional pattern incorrectly classified (which is equivalent to 99.25% of the patterns classified correctly). Similarly to the classification performance of ELM within the single classes, ESN algorithm also showed a similar classification accuracy for both classes and was not biased towards either of the classes, with 99.32% of the patterns from the high-class and 99.17% of the low-class being classified correctly. This corresponds to 4 patterns from the high-class and 5 patterns from the low-class being misclassified, out of 9 misclassified patterns in the testing dataset in total.

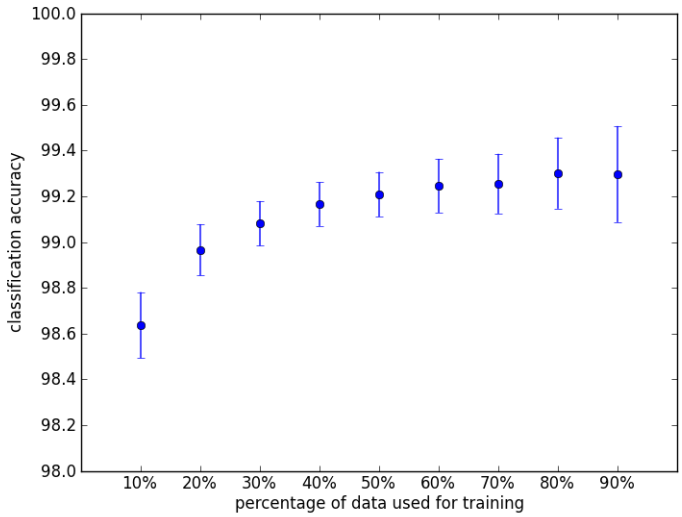

Figure 3: Average classification accuracy $(+/- \sigma)$ of the ELM algorithm on different apportionments between training and testing data

3.4. Results of the case study

The input data for the meta classifiers, p_i^{nn}, p_i^{out} and p_i^{alg}, for both the training and testing datasets, were computed based on the strangeness values α_i^{nn}, α_i^{out} and α_i^{alg}. The number of nearest neighbors, k, used to compute α_i^{nn} and α_i^{out} was set to 10 by trial and error (selecting from the tested data range of [5,100]). Furthermore, the target output y_i^{meta} was computed. After the training phase, the meta algorithm was tested on the testing dataset for both main classifiers.

The outputs of the meta classifier were taken as strangeness values, α_i^{meta}, and the typicalness values p_i^{meta} were computed based on Equation 2.

For the ELM main classifier, the p_i^{meta} for the eight patterns misclassified by the main classification algorithm showed a high degree of untypicalness. The exact values were in the interval [0.0, 0.009]. The typicalness value of 0 means that in the training dataset there were no other patterns with the same degree of randomness.

The upper bound of the interval was taken as the threshold of low reliability of the classification output. Consequently, if the typicalness value of a classified pattern is below 0.009, the classification of the main algorithm is considered as unreliable. In the testing dataset there were three additional patterns with typicalness values below the defined threshold. Therefore, even though these patterns have been classified correctly by the classification algorithm, their classifications are not reliable and in real-world applications decision makers would not trust them.

A similar behavior of the meta classifier was observed for the ESN main classifier. Indeed, as for the ELM classifier, all of the missclassified patterns showed a high degree of untypicalness. For ESN, the interval of typicalness values was slightly wider [0.0, 0.012]. Taking the upper bound of the interval as the threshold for unreliably classified patterns, results in five additional patterns showing a high degree of untypicalness to those already misclassified by the algorithm.

4. Discussion

In the case study of fault diagnosis in railway turnout systems, the meta classifier was applied to evaluate the reliability of two different main classifiers. The meta classifier was able to detect the difference in the confidence of the two classifiers with a different number of patterns being unreliably classified by the two main classifiers.

Table 1 resumes the main results.

Table 1: Performance of the meta classifier on the output of the two main classifiers

Main algorithm	Number of misclassified patterns in the training dataset	Number of misclassi-fied patterns in the testing dataset	Value range of untypicalness values	Threshold for reliably classified patterns	Number of unreliably classified patterns
ELM	32	8	[0,0.009]	0.009	11
ESN	61	9	[0,0.012]	0.012	14

In the case of the ESN classifier, more patterns were missclassified in the training dataset (61 patterns compared to 32 missclassified by ELM), so that the meta classifier had more negative patterns to generalize. The value range of the untypicalness values for the ESN classifier is wider than that of the ELM classifier and the meta classifier predicts that more patterns are unreliably classified by the ESN classifier than by the ELM classifier (14 compared to 11).

Even though in the considered case study, the meta classifier showed a good performance, it should be mentioned that the meta classifier may also not be able to recognize unreliably classified patterns. If required, the parameters of the algorithm should be adjusted or a different algorithm should be selected. Furthermore, bootstrapping could be applied to more frequently present the patterns misclassified by the main classifier to the meta classifier.

The definition of the threshold is pivotal within the entire evaluation process because a threshold that is too small would miss misclassified patterns and a too large threshold would result in too many false positive patterns, and this could also be critical in some applications. In this research, the threshold was defined by the upper bound of the untypicalness values of the missclassified patterns by the main classifier. However, alternative approaches can be applied to define the threshold value. Depending on the consequences of false negatives, respectively false positives, the threshold can be adjusted according to the criticality of the application.

The typicalness approach is based on the *iid* assumption, which is a comparably weak assumption. For other approaches, such as, for example, the Bayesian approach, more assumptions are, usually, required. However, the proposed approach is not applicable if this assumption does not hold.

Typicalness values are relative values, which makes the approach flexibly applicable and the results comparable. However, for datasets with a very small value range and small deviations between the values, the typicalness values may result in misleading conclusions. Therefore in these cases, not only the typicalness values have to be assessed but also the strangeness values, which can provide complimentary information.

5. Conclusions

This paper proposes a meta approach to quantify the reliability of classifiers of individual patterns of data collected for fault diagnostics purposes. The approach comprises two classification steps: a classifier performing the main classification task and a meta classifier. The meta classifier uses typicalness values as input derived from the input and the output of the main classifier. The reliability evaluation of the main classifier is based on the typicalness values computed based on the output of the meta classifier. With this, a threshold for unreliable classifications is defined to identify patterns that have been unreliably classified by the main algorithm.

The proposed approach was applied to fault diagnosis in railway turnout systems, showing good performance in quantifying the reliability of two main classifiers (echo state networks and extreme learning machines). Further research will be required to validate the proposed approach more thoroughly and to demonstrate its specific advantages and limitations by comparison to other methods of classification. For this validation, either a dataset covering all the specific characteristics, such as the inclusion of novel patterns, will be required or specifically designed synthetic datasets will be constructed.

6. Acknowledgments

The authors would like to thank BLS AG for providing the data for this research project.

The participation of Olga Fink to this research is partially supported by the Swiss National Science Foundation (SNF) under Grant No. 205121_147175.

The participation of Enrico Zio to this research is partially supported by the China NSFC under Grant No. 71231001.

7. References

[1] E. Alpaydin, Introduction to machine learning, 2nd ed., MIT Press, Cambridge, Mass., 2010.
[2] P. Baraldi, R. Razavi-Far, E. Zio, A method for estimating the confidence in the identification of nuclear transients by a bagged ensemble of fcm classifiers, in: Seventh American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies NPIC&HMIT, American Nuclear Society, 2010, pp. 283–293.
[3] D. Barbara, C. Domeniconi, J.P. Rogers, Detecting outliers using transduction and statistical testing, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2006, pp. 55–64.
[4] Y. Bengio, Gradient-based optimization of hyperparameters, Neural Computation 12 (2000) 1889–1900.
[5] Z. Bosnic, I. Kononenko, An overview of advances in reliability estimation of individual predictions in machine learning, Intelligent Data Analysis 13 (2009) 385–401.
[6] B. Chacko, V. Vimal Krishnan, G. Raju, P. Babu Anto, Handwritten character recognition using wavelet energy and extreme learning machine, International Journal of Machine Learning and Cybernetics 3 (2012) 149–161. URL: http://dx.doi.org/10.1007/s13042-011-0049-5. doi:10.1007/s13042-011-0049-5.
[7] M. El-Koujok, M. Benammar, N. Meskin, M. Al-Naemi, R. Langari, Multiple sensor fault diagnosis by evolving data-driven approach, Information Sciences (2013).
[8] A. Gammerman, V. Vovk, Prediction algorithms and confidence measures based on algorithmic randomness theory, Theoretical Computer Science 287 (2002) 209–217.
[9] S.S. Haykin, Neural networks and learning machines, 3rd ed., Pearson Education, Upper Saddle River, 2009.
[10] M. Hermans, B. Schrauwen, Recurrent kernel machines: Computing with infinite echo state networks, Neural Computation 24 (2012) 104–133.
[11] T. Heskes, Practical confidence and prediction intervals, Advances in neural information processing systems (1997) 176–182.
[12] A.E. Hoerl, R.W. Kennard, Ridge regression: Biased estimation for nonorthogonal problems, Technometrics 12 (1970) 55–67.
[13] G.B. Huang, D. Wang, Y. Lan, Extreme learning machines: a survey, International Journal of Machine Learning and Cybernetics 2 (2011) 107–122.
[14] G.B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 42 (2012) 513–529.
[15] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: Theory and applications, Neurocomputing 70 (2006) 489–501.
[16] H. Jaeger, H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, Science 304 (2004) 78–80.
[17] N. Japkowicz, M. Shah, Evaluating learning algorithms a classification perspective, Cambridge University Press, Cambridge, 2011.
[18] A.K.S. Jardine, D. Lin, D. Banjevic, A review on machinery diagnostics and prognostics implementing condition-based maintenance, Mechanical Systems and Signal Processing 20 (2006) 1483–1510.
[19] I. Kononenko, I. Bratko, Information-based evaluation criterion for classifier's performance, Machine Learning 6 (1991) 67–80.
[20] Y. Lan, Y.C. Soh, G.B. Huang, Ensemble of online sequential extreme learning machine, Neurocomputing 72 (2009) 3391 – 3395. URL: http://www.sciencedirect.com/science/article/pii/S0925231209000782. doi:http://dx.doi.org/10.1016/j.neucom.2009.02.013.
[21] M. Li, P. Vitanyi, An introduction to Kolmogorov complexity and its applications, Springer, New York [etc.], 1993.
[22] M. Lukosevicius, A practical guide to applying echo state networks, A Practical Guide to Applying Echo State Networks, volume 7700 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2012, pp. 659–686.
[23] M.S. Mahmoud, H.M. Khalid, Expectation maximization approach to data-based fault diagnostics, Information Sciences 235 (2013) 80–96.
[24] G. Manjunath, H. Jaeger, Echo state property linked to an input: Exploring a fundamental characteristic of recurrent neural networks, Neural computation 25 (2013) 671–696.
[25] P. Martin-Löf, The definition of random sequences, Information and Control 9 (1966) 602 – 619. URL: http://www.sciencedirect.com/science/article/pii/S0019995866800189. doi:http://dx.doi.org/10.1016/S0019-9958(66)80018-9.
[26] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, A. Lendasse, Op-elm: Optimally pruned extreme learning machine, Neural Networks, IEEE Transactions on 21 (2010) 158–162. doi:10.1109/TNN.2009.2036259.
[27] S.M. Omohundro, Five balltree construction algorithms, International Computer Science Institute Berkeley, 1989.
[28] Z.N. Sadough Vanini, K. Khorasani, N. Meskin, Fault detection and isolation of a dual spool gas turbine engine using dynamic neural networks and multiple model approach, Information Sciences (2013).
[29] E. Smirnov, S. Vanderlooy, I. Sprinkhuizen-Kuyper, Meta-typicalness approach to reliable classification, Frontiers in Artificial Intelligence and Applications 141 (2006) 811.
[30] G. Vachtsevanos, Intelligent fault diagnosis and prognosis for engineering systems, Wiley, Hoboken, NJ, 2006.
[31] V. Vovk, A. Gammerman, C. Saunders, Machine-learning applications of algorithmic randomness, in: Proceedings of the Sixteenth International Conference on Machine Learning (ICML-1999), Morgan Kaufmanns, 1999, pp. 444–453.
[32] Q. Wu, R. Law, Complex system fault diagnosis based on a fuzzy robust wavelet support vector classifier and an adaptive gaussian particle swarm optimization, Information Sciences 180 (2010) 4514–4528.